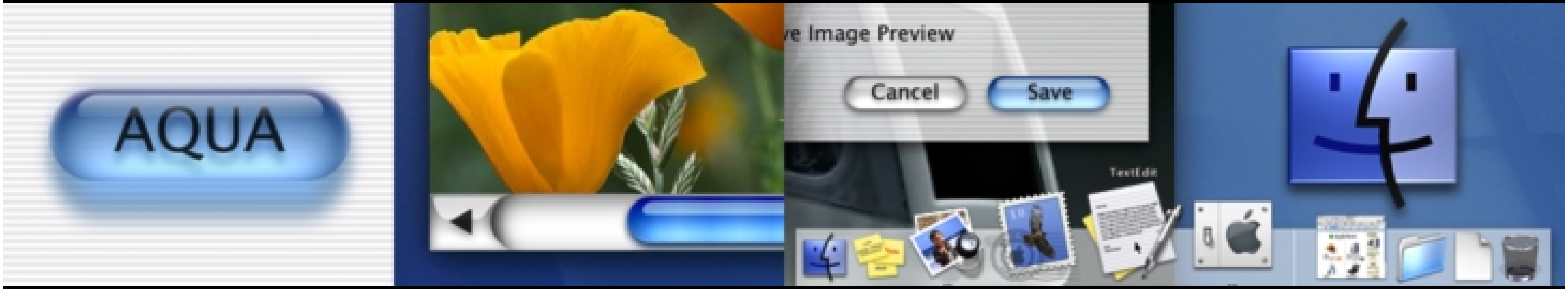
Session 408

Converting Objective-C Applications to Java



Francois Jouaux, WebObjects Deployment Manager
Ben Trumbull, WebObjects Engineer

Introduction

- This session
  - Assumes some knowledge of WebObjects
  - Covers strategies for converting ObjC, WebScript and Bridged Java projects
  - Shows our supplied conversion tools

What You’ll Learn

- The goals of pure Java in WebObjects 5
- Who should convert ?
- Who should not ?
- API and feature changes in WebObjects pure Java
- The conversion process
- Hints on how to extend the conversion

What Is the WebObjects 5 for Java Release ?

- WebObjects 4.5 Ported to Java
  - Full featured and API compliant with WebObjects 4.5 (WOF, EOF)
  - Works as an add-on to WebObjects 4.5
  - Uses WebObjects 4.5 developer tools
  - Monitoring tools are compatible with WebObjects 4.5
  - A conversion tool for WebObjects applications

Benefits of the Pure Java Release I

- Development on Mac OS X, Mac OS X Server, and Windows NT/2000
- Deployment on above platforms, plus Solaris, HP, and LINUX
- Smaller runtime footprint opens the door to more platforms

Benefits II: Better Integration

- Better Java Experience
- JDK 1.1.8 based and
  JDK 1.2 and 1.3 compliant
- Allows for third-party development tools, debuggers and optimizers—impossible with Bridged Java
- Integrates with any third-party products with a Java interface

Benefits III:
More Database Connectivity

- Extended JDBC support
- JDBC plug-in technology allows for easy customization of the JDBC interface
(Oracle, Sybase , Informix, OpenBase provided)
- LDAP adaptor

Who Should Convert ?

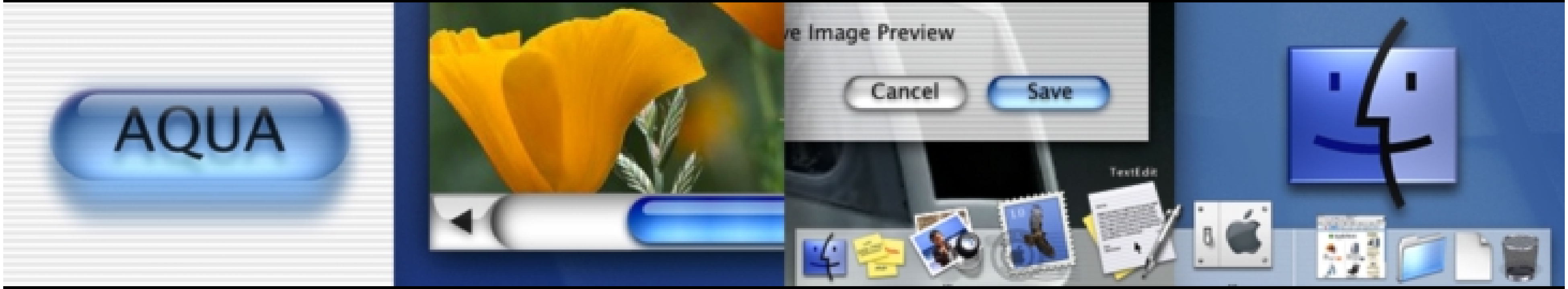
- Objective-C or WebScript developers requiring more deployment solutions
- Existing Bridged Java developers
- Anyone requiring access to more Java standards

Who Should Think Twice Before Converting ?

- If your WebObjects Applications or frameworks…
  - Contain tons of categories on our NS*, WO* or EO* objects
  - Rely heavily on the poseAs: feature
  - Use low-level Objective-C runtime found in the System.framework/Headers/objc header files
  - Use low-level C function calls or structures
  - Are in production and do not need a major update
- …then you should not convert your project

The Conversion Process

DEMO

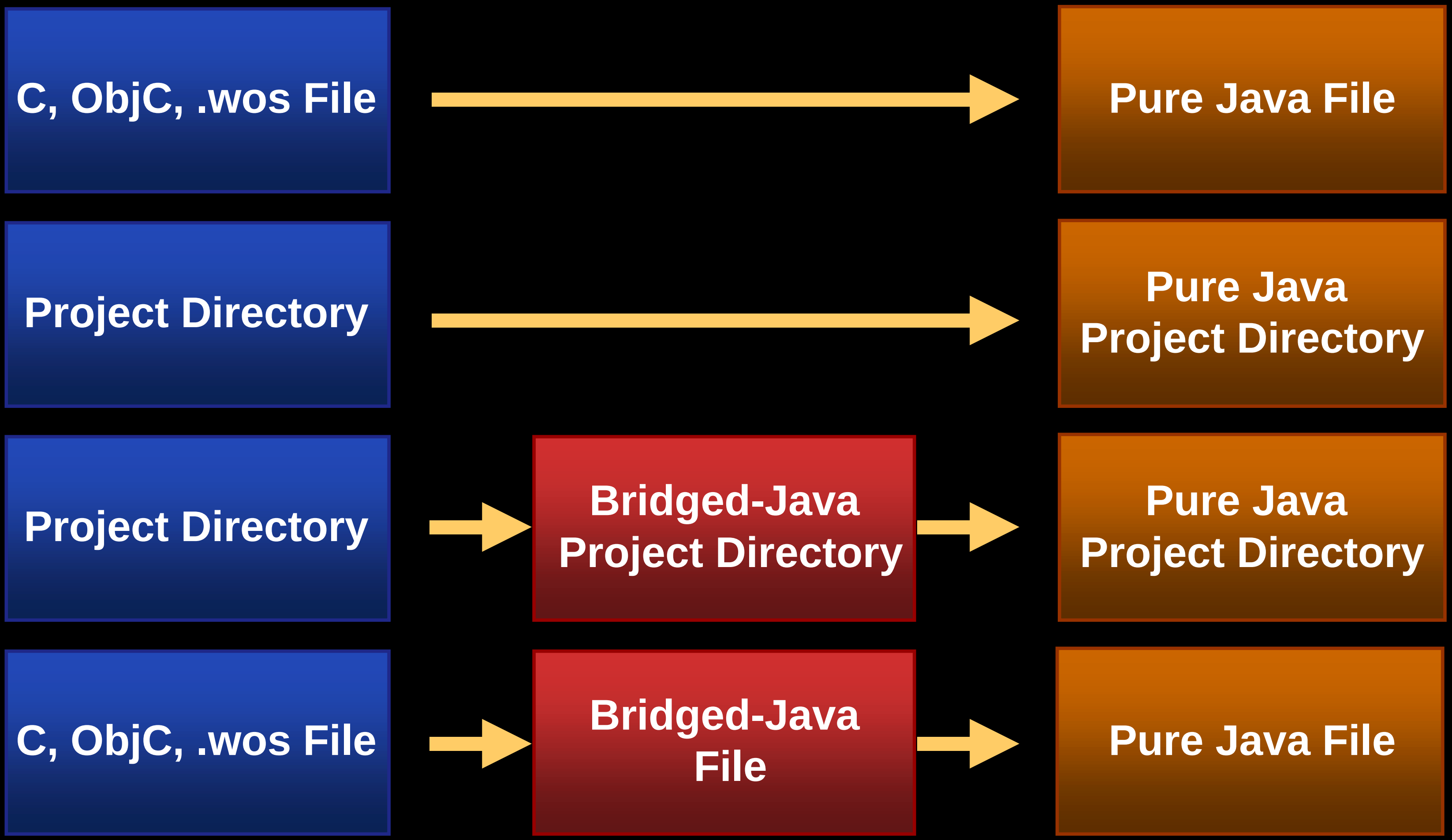


A Simple Example Conversion

JavaConverter

- Project-wide conversion mode
- File-by-file conversion mode for touch-up work
- “All-at-once” or “Bridged-Java only” conversion modes

JavaConverter Steps



Convert in One Step

- For smaller projects
- Expect a long period of non-running project
- Objective-C, WebScript, Bridged-Java files are converted
- Resources (WOComponents or models), makefiles and framework references are corrected

Two Steps Conversion

- “Bridged-Java Only” mode converts your project and ObjC/WOS code files to complete Bridged-Java
  - Recommended path for large projects
  - Allows for incremental framework conversion
  - Down to class-by-class conversion (requires wrapping remaining ObjC classes)
- Once running, project can be converted to pure Java by using the JavaConverter again

Resulting Pure Java Code

- Code organization and all comments are preserved
- Source is “beautified” to follow coding standards
- Three levels of conversion errors are easily searchable in the produced Java code

Compile Time

- Once errors have been corrected, time to compile
- You’re 80% there!
- Takes the hassle out of converting
- Human translation rate for Objective-C to Java is about 300 lines/day
- JavaConverter does the same in 10 seconds and is much more reliable

Classes Issues

- Some Foundation classes are not available anymore
- Classes with no meaning in Java (ex. NSAutoreleasePool) or with a Java equivalent (ex. NSThread)
- NS* structs are gone, except NSRange

Categories Issues

- Categories on classes in the same project are reunited
- Categories on external classes are transformed in helper classes
- BUT: new helper classes cannot be referenced where needed automatically

Macros and Functions Issues

- All macros are resolved (may need to repeat conversion on all platforms)
- Exception handling is translated
- All NSFunctions() are gone
- Many are automatically replaced with their equivalent
- C Function calls must be corrected manually

API Issues

- Most of our API changes are automatically done for you
- Our classes have kept their Bridged API
- API changes in your frameworks can be automated

Hint I: Objective-C Conversion

- It works like an Objective-C compiler
- But not a good C compiler
- The grammar is provided
- Uses cpp (macro preprocessing)
  and javacc (from Metamata)

Hint II: WOS Conversion

- Extended WebScript compiler
- Generates pseudo ObjC (won’t compile)
- The ObjC grammar is loose enough to continue the conversion to Java

Hint III: Bridged-Java Conversion

- This step takes Bridged-Java code to WebObjects 5 for Java
- Handled with tops scripts
- All imports are corrected
- All API changes are applied

Extensible

- Source code and grammar are provided
- Just drop scripts in scripts folders to improve pre- and post-conversion steps
- You can automate your framework API changes

Conclusions: What We Covered Today

- The WebObjects 5 environment
- JavaConverter: A great tool that handles much uninteresting work
- Try it out in the upcoming Beta

For More Information

http://www.apple.com/webobjects

Visit the WebObjects lab downstairs!
Everyday from 11:00 a.m.–2:00 p.m.

Try out your WebObjects 4.5 Evaluation CD!

WebObjects Community BOF
Wed., 6:30 p.m.–8:00 p.m.

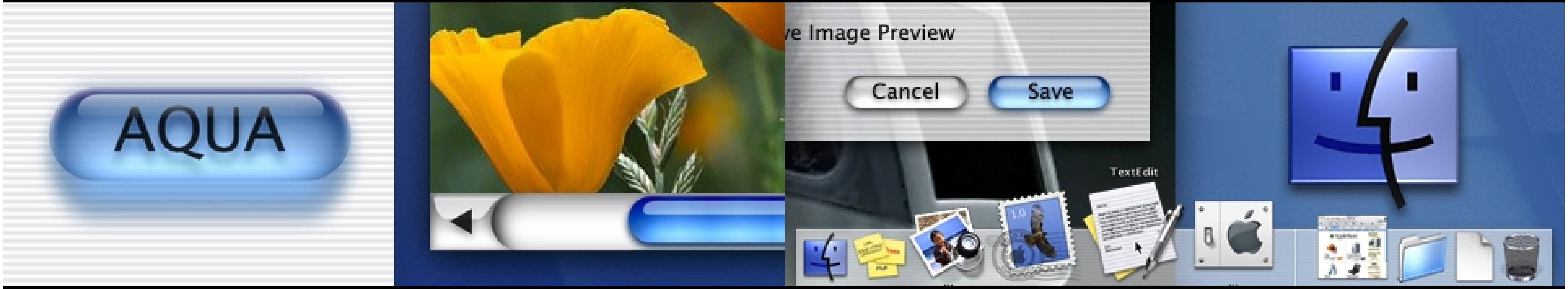
Think different.

Think different. . .

About the conversion process.

Session 408

Q&A



Francois Jouaux, WebObjects Deployment Manager
Ben Trumbull, WebObjects Engineer

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
wofeedback@group.apple.com

Ernest Prabhakar
Product Line Manager, WebObjects
webobjects@group.apple.com

WWDC

Worldwide Developers Conference 2000

Think different.